PROCEEDINGS OF SCIENCE

3.5 Million Smartmeters in the Cloud*

Aleš Křenek

E-mail: ljocha@ics.muni.cz

Petr Holub

Masaryk University

Vlastimil Holer

Masaryk University

Daniel Kouřil†

Masaryk University

Filip Procházka

Mycroft Mind

Zdenek Hejna

Mycroft Mind

Miroslav Guričan

Mycroft Mind

František Muller

ČEZ

Currently the operator of the Czech national electric power distribution network runs a pilot project of testing deployment of approximately 35,000 smart meters – devices which measure various quantities related to power consumption at the end-user side. The measured data are gathered along the distribution to concentrator devices (each serving several hundreds of smart meters typically) which send it further to a cluster of central servers for data processing.

The pilot will be followed by deployment of approximately 100x more such devices. However, such deployment is unprecedented with the software in use. Therefore a simulation experiment is required to discover and mitigate potential scaling issues in advance.

The simulation is run in the cloud infrastructure of CERIT-SC (managed by the OpenNebula stack). The central servers are deployed on hardware of similar scale and in the same software configuration as intended for the production. The data sources – smart meters are simulated at the level of concentrators. The implementation allows to simulate varying patterns of measurements and patterns of unaccessibility. Unchanged communication protocols between the concentrator and the central server are used then. Because the target setup assumes up to 50,000 concentrators, up to thousands of concentrators must be handled by a single simulator instance.

The target environment uses data links of varying qualities, starting even at GPRS links. Therefore the simulator also includes modules which emulate packet loss, delay, and jitter in the network communication in a controlled way. This setup allows testing sensitivity of the entire system on network properties.

The cloud is an essential flexible infrastructure to run such an experiment. With the exception of the RDBMS server running on a large SMP machine all the servers (both the central cluster and the data source simulators) are run in virtual machines in the cloud. This allows multiplatform setup (Windows and two Linux flavours are used) and, in particular, servers can be added to various groups (simulators, front-ends, application logic, etc.) easily. On the other hand, resources which are not required in a certain stage of the experiment, can be easily released to other use, which makes even such a large experiment affordable.

The International Symposium on Grids and Clouds (ISGC) 2013
17 – 22 March 2013
Academia Sinica, Taipei, Taiwan

[*] This work was done, and hardware resources were provided with the support of the project CERIT Scientific Cloud, part of the Operational Program Research and Development for Innovations, reg. no. CZ. 1.05/3.2.00/08.0144. The work also leveraged the cloud environment build as a joint work with the National Grid Infrastructure MetaCentrum, provided under the programme “Projects of Large Infrastructure for Research, Development, and Innovations” (LM2010005).

[†] Speaker.

1. The Usecase

1.1 Pilot Smartmeter Deployment in the Czech Republic

CEZ company runs a smart metering pilot project and installed 35,000 smart meters in its energy grid (it is approximately 1% of total metering points in the CEZ network). Four different location were selected (different network structure and age, demography, weather conditions), five types of smart meters are used, three alternatives of data center for collecting and storing measured data are used. The tested system is fully integrated with legacy systems. The main goal is to evaluate the used technology and to learn how such complex system behave in different conditions and situations.

It might seem that 35,000 smart meters (or even 3,500,000 expected in full roll-out) is not such a big number in comparison with large smart meter installations in France or USA (tens of millions [1]). But requirements of CEZ are much higher. A solution which reads energy consumption from smart meter once per day (or week) cannot be used for controlling some aspects of the grid. The Czech energy system is already quite sophisticated. For example, more than 50 years the ripple control technology for optimizing balance between energy production and energy consumption is used. This technology is mainly used for minimizing between daily and nightly consumption. Useful solution for CEZ have to collect data perpetualy (every 15 minutes) and also have to offer the possibility to send messages and commands to individual smart meters quickly and reliably.

1.2 Proposed Target Deployment

The target deployment will be a communication and control network containing the following layers:

1. Communication with invididual smart meters is processed by data concentrators. Communication line between smart meters and data concentrator is available typically using PLC (Power Line Comunication) or BPL (Broadband over Power Lines). However, these communication channels are very noisy and their capacity is limited.

2. Communication between data concentrators and data center is typically GPRS (where problems with line speed and outages also appear).

The proposed solution have to be very precisely designed to handle these restrictions. It contains selection of communication protocols, selection of communication modes (push or pull) and configuration of system parameters.

The proposed target deployment have to provide the following functionalities:

- to be able to get most of consumption measurements from all data concentrators in 15-minutes refresh rate

- to use smart meters for switching on or off specific consumption segments and to manage consumption actively

1.3 Simulation Goals

The main goal of the simulation is testing several alternative solutions for collecting and storing data from smart meters and answering the following questions for each alternative:

- Is the tested alternative able to process 3,500,000 smart meters?
- How many computational resources the tested alternative needs to process data from smart meters?
- What type of problems are characteristic for tested alternative?

Answering these questions will help to find parameters of optimal architecture which will be able to work effectively and safely.

On the other hand, the whole experiment is particularly challenging for the CERIT-SC cloud infrastructure, as a unique opportunity to evaluate flexibility of the environment using a real-life scenario, as well as to discover potential specific problems (technical, psychological, legal, ...) of collaboration between the academia and the industry. We focus on the following issues:

- *Interference with the other users.* The whole experiment is large enough and it will last for non-negligible time when it will consume considerable portion of the hardware resources of the centre. What will be the impact on the other users, and how will they react (if they even notice)?
- *Identity management and policies.* People coming from industry, who are not eligible to become users of the centre if existing policies are applied strictly, will need user accounts and will have to access the resources. How will they be registered, what guarantees will be given in order to keep the resource usage restricted to this experiment only, and what means of isolation to interfere with other users will be applied?
- *Testbed setup and maintenance.* The testbed consists of two large SMP machines (80 cores), which are available for the experiment as dedicated physical hosts, and a large number of smaller (12 cores) which serve the cloud infrastructure. What issues will emerge from the split of responsibilities to install and maintain sofware on the physical machines? How virtual machine images will be prepared, distributed, shared and replicated? What versioning mechanisms will be applied?
- *Security and administrator's responsibility.* By setting up the experiment we create co-administered environment with direct involvement of both industrial and academic partners. How the responsibilities for keeping the environment secure, and resovling security incidents eventually will be split? On the other hand, how the intelectual property of the industrial partner will be protected?

The experience from the experiment gives at least partial answers to all these questions.

2. Simulation Environment

2.1 The CERIT-SC Cloud Infrastructure

At the time of writing this paper, the CERIT-SC centre was approximately in the middle of its building. The available hardware consists of 20 SMP machines of 80 cores (Intel Xeon E7-2860) and a small cluster of 48 machines with 12 cores (Intel Xeon E5-2620). All the machines run Debian 6 operating system currently. Further details can be found at `http://www.cerit-sc.cz/`.

All the 12-core machines are exposed to the users in the cloud environment. The mainstream KVM virtualization [2] layer is used to provide the hosting environment. OpenNebula [6] is used to manage the whole cloud environment. The cloud environment[1] was developed in close collaboration with the Czech NGI Metacentrum.

2.2 Identity Management

CERIT-SC (and Metacentrum as well) use the Perun [3] identity management system, which is designed to import user identities from various sources (human-resource databases of universities in particular).

The system, in order to deal with very heterogeneous environment in terms of level of trust of the user authentication, policies to access resources etc., is build around the concept of *Virtual organization* (VO). Any VO, which spans multiple real-world organizations typically, can define which user identity sources it accepts, what are its policies, and it negotiates contracts to access resources (computing, storage, software, . . .).

We decided to set up a dedicated VO for the purpose of this experiment. A separate VO allows to relax, w.r.t. the standard restrictions of the acedemic environment, the policy defining eligible users, so that it is open to the industrial partners. On the other hand, specific statements of the policy bind the users to use the resources exclusively for the purpose of this collaborative experiment.

2.3 Setup Overview

The computing infrastructure used for the experiment consists of the following components:

- Medium-sized SMP machines (80 CPU cores) were used as the database servers of the data centre software. The servers were chosen due to large memory and disk space requirements. Oracle Linux and Oracle database were used. These machines are (due to other technical reasons) not included in the CERIT-SC OpenNebula cloud, and they were provided in dedicate mode.

- Components of the data centre (web interface, communication servers etc.) were run in the cloud environment (12-core virtualized machines). The installations were prepared as contextualizable images (Windows typically) with all the software installed, and they were instantiated in the required number of copies.

[1] `https://wiki.metacentrum.cz/wiki/MetaCloud_testbed_-_service_description`

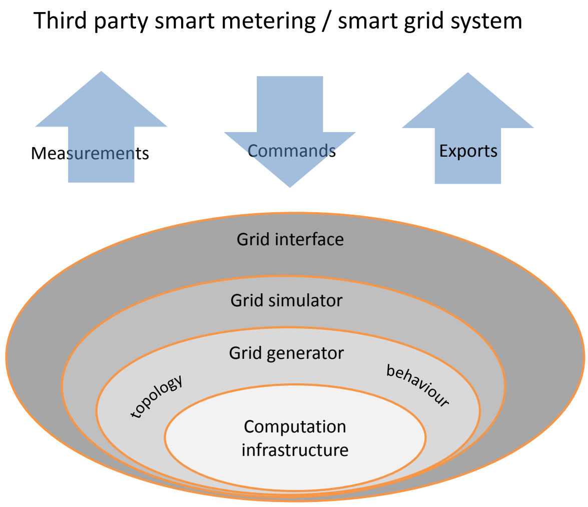

Figure 1: Simulator schema

- Simulator of the data concentrators. Similarly to the data center components, virtual machine images were prepared and instantiated in the (varying) required number of copies. Standard Linux images of CERTIT-SC were used as the baseline.

- Dual network interfaces of the cloud machines were leverage – one for the actual simulated traffic, where the NetEm traffic shaping was applied (Sect. 2.5), and the other for control of the components, in order to prevent both interference of the control and monitoring traffic with the experiment, and to avoid annoying effects of the traffic shaping on the control connections.

The flexibility of the cloud environment was highly leveraged. As the demand of the experiment evolved over time (starting with simulating smaller number of smart meters, and overlapping different setups of the data centre), required numbers of the system components could be allocated, while making all the other resources still available to other users.

2.4 Simulation of low voltage energy grid

The simulator generates a flow of measurements from the simulated low voltage energy grid. Structure of the grid is generated according to specified parameters using the grid topology generator. Individual smart meters and data concentrators behave according to a simulation plan generated by the grid behaviour generator. Interaction between the simulated grid and a tested third party solution is available using the grid interface. The grid interface offers several protocols used in a domain of smart metering (for example DLMS, AGS-DK) for on-line communication. The grid interface also offers possibility to export static or dynamic data from grid (typically in a form of CSV file). The simulator is composed from following parts:

Grid topology generator creates a low voltage energy grid according to specified parameters.

The most important parameters are:

- *number of transformer station* – each transformer station has it own data concentrator which collects information from smart meters. For example the simulated grid of CEZ has approximately 50,000 transformer stations.
- *numbers of smart meters at transformer stations* – this parameter allows to create categories of transformer stations according their size (for example, a small one which has between 1 to 20 smart meters) and for each category define proportionally their occurrence in the grid (for example, the grid will contain 20% of small transformer stations).
- *consumption patterns* – allows to define daily consumption patterns (96 values – for each 15 minutes) and their distribution in the grid
- *tariffication distribution* – allows to specify tariffication categories (for example one tarif consumption, two tarif consumption) and their distribution in the grid
- *communication line quality* – allows to create categories of data concentrators and to specify quality of communication line (for example packet rate, packet loss, latency, jitter)

Grid behaviour generator generates simulation plans. A simulation plan contains information what data will be send (in a push mode) or available (in a pull mode) from which smart meters and data concentrators. Simulation plan is then interpreted by the grid simulator. The simulation plan allows to simulate accessibility of data from smart meters and data concentrators. There can be simulated situations where there are some communication failures and part of data are not accessible or communication with some smart meters is hard, and data are available only few times per day.

The most important parameters are:

- *type of communication* – specifies if measurements will be pushed from concentrator or if the third party solution will ask simulated data concentrators for measurements
- *scenario length* – number of scenario steps (typical scenario for CEZ contains 288 steps – three days, one step every 15 minutes)
- *time density* – delay between steps (typically 15 minutes)
- *data concentrators accessibility distribution* – allows to create categories of data concentrators according to their accessibility (for example badly accessible data concentrators will be accessible only 10% time of day) and to define proportion of category occurrence (for example the grid will contain 30% of badly accessible data concentrators).
- *smart meters accessibility distribution* – allows to create categories of smart meters according their accessibility (for example badly accessible smart meters will be accessible only 5% time of day) and to define proportion of category occurrence (for example the grid will contain 20% of badly accessible smart meters).
- *timestamp of the first measurements* – timestamp of the first simulation step

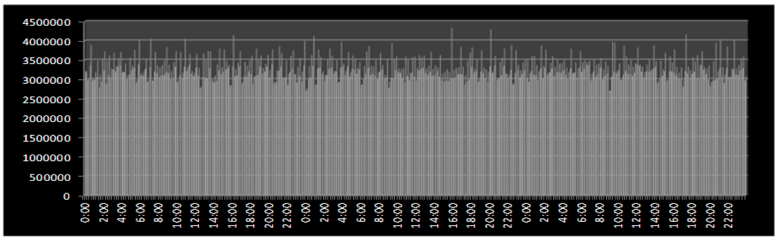

Figure 2: Number of communicating meters during test

Grid interface allows to communicate with the simulated grid by protocols used in smart metering domain (namely DLMS and AGS-DK). These protocols define structure of messages for querying or sending measurements.

2.5 Network Throttling

In order to simulate varying network conditions between the concentrators and central servers, we used the *netem* framework [4] to set the following parameters:

- *per-connection bandwidth limit* implemented indirectly using *hasthable* functionality, set to 15 packets per second limit with 5 and 1 packet bursts for TCP and other traffic respectively;
- *network latency* with normal distribution and the following settings (the second number after ± defines width of the distribution): 600 ± 200 ms, 800 ± 400 ms, 1000 ± 500, $1400 \pm 600 ms$ for 4 classes of different clients;
- *packet loss* between 0.1%–0.4% with no correlation.

These parameters we found to be reasonable approximations of the mobile links, within the practical limits of the simulation. The netem was setup on the server side of the simulator and was in place for both inbound and outbound traffic.

2.6 Testing Scenarios

We derived parameters of the pilot project smart metering infrastructure and we generated two version of the simulated grid – a small grid containing 100,000 smart meters and a large grid containing 3,500,000 smart meters. We also derived patterns of smart meters and data concentrators accessibility to be able to simulate problems in PLC/BPL/GPRS communication. Several scenarios were tested – a typical test was 3 days long – it means that more than billion measurements were sent during tests on the large grid.

2.7 Monitoring

The monitoring helps to collect all relevant data for analyzing test scenario results. It is necessary to collect information from every layer of the environment – usage of hardware resources as CPU or memory usage, network traffic, database performance or events from application logs. It is important to evaluate WHAT every component is doing and HOW much it uses resources.

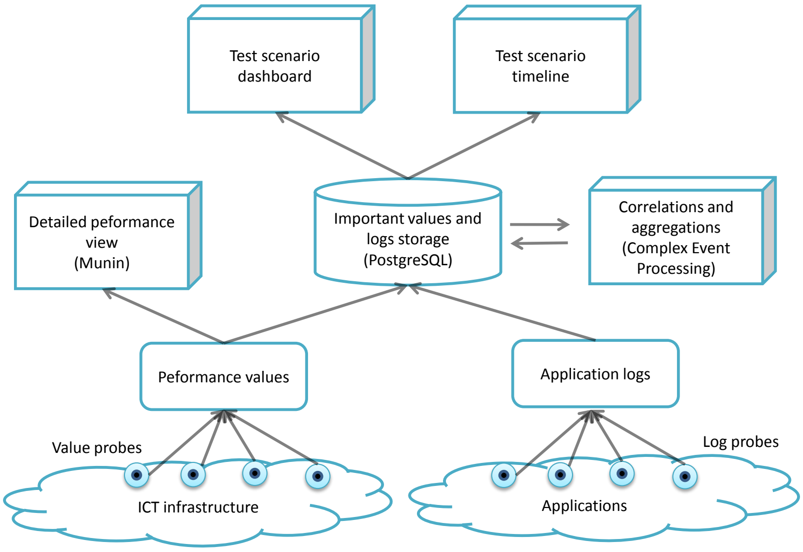
Test scenario dashboard
Test scenario timeline
Detailed peformance view (Munin)
Important values and logs storage (PostgreSQL)
Correlations and aggregations (Complex Event Processing)
Peformance values
Application logs
Value probes
Log probes
ICT infrastructure
Applications

Figure 3: Monitoring schema

The monitoring collects values (for example CPU usage) and events (for example application log records about processed measurements). Values are collected using the Munin system [5] using value probes installed on monitored servers. Events are collected using event probes installed on monitored servers and periodically collecting relevant log records. The collected values are stored at the database. The monitoring system collects millions of records during the tests. It is necessary to aggregate these pieces of information into meaningful indicators and events which help to understand what is happening. Aggregated information are presented in dashboard and timeline.

3. Results and lessons learned

The experiment proved that a similar environment can be successfully used to run a realistic simulation of this type of technology and at this extent. All the tested technologies for data collecting were able to process data from 3,500,000 smart meters in a mode with buffering measurements before storing them to the database. It is possible to simulate large grids on the granularity of an individual smart meter. The simulator is a tool which can be used for testing architectures of smart metering systems and third party solutions evaluation or tuning. The results of the payload tests provide valuable feedback to the Czech power network operator CEZ.

OpenNebula was chosen for its simple deployment, common hypervisors support, rich CLI tooling and wide configuration options for each virtual machine. It can be used in nearly default settings for most scenarios, but also covers edge cases like optimized NUMA-aware HPC. CERIT-SC's cloud infrastructure offers full virtualization (HVM) via Linux KVM hypervisor and QEMU emulator. This environment proved to be fully functional for this kind of experiments.

No special optimization was required on the hypervisor side (for example no NUMA, no VCPU-CPU binding). Paravirtualized drivers (virtio) for disk and network I/O were used to archieve higher performance both in Linux and Windows virtual machines.

Virtual machine images used size-growing QCOW2 format. Initial optimal file size allowed us quick deployments, while preserving still good performance in compare with raw image or even direct block access. Windows images were originally prepared on VMware in VMDK disk format. These were later converted into QCOW2 to improve the overall performance in the QEMU emulator.

Minor problems occured with Linux firewall and its connection tracking functionality on physical hosts with virtual machines used as a central points in tests. With high number of concurrent connections from many virtual machines, the fixed size connection tracking table was quickly filled. This resulted in network traffic drop-outs. The problem was fixed by increasing the table size (`net.nf_conntrack_max`).

Special care must have been paid to the configuration of Windows background services (for example Windows Update) which may generate unpredictable background load on both CPU and network and they may bias the test results considerably.

Due to good isolation of the experiment in the cloud environment, no particular interferences with other users were observed.

During the experiment we faced a minor security incident when a forgotten install-time default password to a service was exploited by a remote attacker. However, the exploit was detected by the standard security mechanisms of CERIT-SC and administrator of the affected virtual machine alerted, and he resolved the incident immediately.

Altogether, the experiment serves as a good example of collaboration between academia and industry, and it is a pilot for further such collaborations. The main benefit for the industrial partner is testing its technology in a scale that would be difficult (or unaffordably expensive) to achieve otherwise. On the other hand, the academic partner gains valuable experience in the of the resources in this scale and mode, pushing its infrastructure-related research further.

References

[1] Utility-scale smart meter deployments, plans, & proposals. IEE Report, May 2012.

[2] Avi Kivity et al. kvm: the Linux virtual machine monitor. In *Proc. the Linux Symposium*, 2007.

[3] Michal Procházka et al. Perun – user and resource management system. `http://perun.metacentrum.cz/web/overview.shtml`, 2013.

[4] S. Hemminger. Network emulation with NetEm. In *Proc. the 6th Australia National Linux Conference (LCA2005)*, April 2005.

[5] Patricia Jung. Munin–the raven reports. *Linux Journal*, 180, 2009.

[6] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente. IaaS cloud architecture: From virtualized datacenters to federated cloud infrastructures. *IEEE Computer*, 45:65–72, Dec 2012.